What is claimed is:

1. A method of preparing metadata for splicing of a transport stream including video access units encoding video presentation units representing video frames, the video access units of the transport stream encoding the video presentation units using a data compression technique and containing a variable amount of compressed video data, the method including:

a) a server ingesting the transport stream;

b) the server storing the transport stream in a file in data storage;

c) concurrently with storing the transport stream in the file in data storage, the server computing metadata for splicing of the transport stream, and storing the metadata for splicing in the file.

2. The method as claimed in claim 1, wherein the transport stream is MPEG-2 compliant.

3. The method as claimed in claim 1, wherein the ingestion of the transport stream, the computation of the metadata, and the storage of the transport stream and the metadata in the file occurs at least as fast as real time.

4. The method as claimed in claim 1, wherein the data is ingested in the server and the metadata are computed during a metered file transfer protocol (FTP).

5. The method as claimed in claim 1, which includes storing the ingested transport stream in a carousel of buffers, accessing the carousel of buffers to compute the metadata and to store the metadata in the carousel of buffers, and then accessing the carousel of buffers to transfer data of the transport stream and corresponding metadata to the file in data storage.

6. The method as claimed in claim 5, wherein the server includes a stream server computer and a cached disk storage subsystem, the carousel of buffers are allocated in random access memory of the stream server computer, and the stream server computer is programmed with a file system that maps the file to disk storage in the cached disk storage subsystem and that writes the data of the transport stream and corresponding metadata to the file in disk storage of the cached disk storage subsystem.

7. The method as claimed in claim 5, wherein the server includes a network interface board that ingests the transport stream and loads the transport stream into the carousel of buffers using a direct memory access protocol.

8. The method as claimed in claim 1, wherein the computing of the metadata for the splicing of the transport stream includes computing an extrapolated program counter value ($PCR_e$) for a respective first I-frame in each of a plurality groups of pictures (GOPs) in the transport stream.

9. The method as claimed in claim 8, wherein the computing of the metadata for splicing includes computing a decode time stamp (DTS) corresponding to the extrapolated program counter value ($PCR_e$) for the respective first I-frame in each of a plurality of groups of pictures (GOPs) in the transport stream.

10. The method as claimed in claim 9, wherein the respective DTS and $PCR_e$ values for the GOPs are stored in a GOP index in a header of the file.

11. The method as claimed in claim 10, wherein the GOP index further includes at least one frame number and a pointer to the transport stream data in the file for each of said plurality of groups of pictures (GOPs) in the transport stream.

12. The method as claimed in claim 10, wherein the metadata includes values for attributes of each of a plurality of groups of pictures (GOPs) in the transport stream, and the values are stored in a GOP index in the file.

13. The method as claimed in claim 12, wherein the GOP index includes an entry for each of the plurality of GOPs, and each entry includes at least one frame number of a frame in the respective GOP, a pointer to where transport stream data of the respective GOP is stored in the file, and values for other attributes of the respective GOP.

14. The method as claimed in claim 13, wherein the GOP index is in the form of a table and is stored in a header of the file after metadata about the transport stream as a whole.

15. The method as claimed in claim 1, which includes decimating the GOP index by reducing the number of entries in the GOP index to make room for entries of additional GOPs in the transport stream being ingested.

16. The method as claimed in claim 1, which includes skipping metadata computations for a group of pictures (GOP) in the transport stream when there are insufficient computational resources available for computing the metadata for the group of pictures (GOP) concurrently with ingestion of the transport stream.

17. The method as claimed in claim 1, wherein the metadata includes values of attributes of groups of pictures (GOPs) in the transport stream, the attributes include high priority attributes and low priority attributes, and the method includes computing values for both high priority attributes and low priority attributes when there are sufficient computational resources available for computing values for both the high priority attributes and the low priority attributes concurrently with ingestion of the transport stream into the server, and the method includes computing the values for the high priority attributes but not the low priority attributes when there are sufficient computational resources available for computing values for the high priority attributes

but not the low priority attributes concurrently with ingestion of the transport stream into the server.

18. A data storage device containing a file of data of a transport stream including video access units encoding video presentation units representing video frames, the video access units of the transport stream encoding the video presentation units using a data compression technique and containing a variable amount of compressed video data, wherein the file also contains an index to groups of pictures (GOPs) in the transport stream, and the index to the groups of pictures includes pointers to transport stream file data of respective ones of the GOPs, and the file further contains attributes of the GOPs computed from the data of the transport stream, and the attributes of the GOPs are also indexed by the index to the groups of pictures.

19. The data storage device as claimed in claim 20, wherein the index to the groups of pictures is in the form of a table of entries for the respective ones of the GOPs.

20. The data storage device as claimed in claim 19, wherein each entry includes at least one frame number of a frame in the respective GOP, a pointer to where transport stream data of the respective GOP is stored in the file, and values for other attributes of the respective GOP.

21. The data storage device as claimed in claim 18, wherein the index to the groups of pictures is stored in a header of the file after metadata about the transport stream as a whole.

22. The data storage device as claimed in claim 18 wherein the computed attributes for each respective GOP includes an extrapolated program counter value ($PCR_e$) for a respective first I-frame in the respective GOP.

23. The data storage device as claimed in claim 22, wherein the computed attributes for each respective GOP includes a decode time stamp (DTS) corresponding to the extrapolated program counter value ($PCR_e$).

24. The data storage device as claimed in claim 18, wherein the transport stream is MPEG-2 compliant.

25. A method of real-time seamless splicing of a first transport stream to a second transport stream to produce a spliced transport stream, the first transport stream including video access units encoding video presentation units representing video frames, the video access units of the first transport stream encoding the video presentation units using a data compression technique and containing a variable amount of compressed video data, the second transport stream including video access units encoding video presentation units representing video frames, the video access units of the second transport stream encoding the video presentation units using a data compression

technique and containing a variable amount of compressed video data, the first transport stream having a last video frame to be included in the spliced transport stream, and the second transport stream having a first video frame to be included in the spliced transport stream, each of the video access units having a time at which each video access unit is to be received in a video decoder buffer and a time at which said each video access unit is to be removed from the video decoder buffer, said method comprising:

(a) setting the time at which the video access unit for the first video frame of the second transport stream is to be removed from the video decoder buffer to a time following in a decoding sequence next after the time at which the last video access unit for the last frame of the first transport stream is to be removed from the video decoder buffer;

(b) accessing pre-computed metadata for the second transport stream including metadata about a decode time stamp ($DTS_{F2}$) at which the beginning of the video access unit for the first video frame of the second transport stream is removed from the video decoder buffer and an extrapolated program clock reference ($PCR_{e2}$) time at which the beginning of the video access unit for the first video frame of the second transport stream will be received in the video decoder buffer, and using the pre-computed metadata to adjust content of the first transport stream so that the beginning of the video access unit for first video frame of the second transport stream will be received in the video decoder buffer immediately after the end of the video access unit for the last video frame of the first transport stream is received in the video decoder while maintaining the difference ($DTS_{F2}$-$PCR_{e2}$) in the spliced transport stream; and

(c) concatenating a portion of the first transport stream up to and including the last video frame to a portion of the second transport stream including and subsequent to the first video frame.

26. The method as claimed in claim 25, wherein the content of the first transport stream is adjusted by replacing at least one video access unit in the first transport stream with a video access unit encoding a freeze frame having a size selected so that the beginning of the video access unit for the first video frame of the second transport stream will be received in the video decoder buffer immediately after the end of the video access unit of the last video frame of the first transport stream is received in the video decoder buffer.

27. The method as claimed in claim 26, which includes selecting the size of the freeze frame by selecting the size of at least one slice in the freeze frame.

28. The method as claimed in claim 25, wherein the second transport stream has a higher bit transmission rate than the first transport stream.

29. The method as claimed in claim 25, wherein the second transport stream has a lower bit transmission rate than the first transport stream.

30. The method as claimed in claim 25, wherein the first transport stream, the second transport stream, and the spliced transport stream are MPEG-2 compliant.

31. The method as claimed in claim 25, which includes parsing the first transport stream and the second transport streams in real time for audio PES packet headers, parsing the audio PES packets in real time for audio access units (AAUs), identifying in real time non-obsolete AAUs in the first transport stream following the last video frame in the first transport stream and identifying in real time obsolete AAUs in the second transport stream following the first video frame in the second transport stream, reformatting the non-obsolete AAUs in the first transport stream following the last video frame in the first transport stream in real time, eliminating the obsolete AAUs in the second transport stream following the first video frame in the second transport stream in real time, and computing time stamp offsets in real time and re-stamping, in real time, time stamps and continuity counters in the spliced transport stream following the first video frame from the second transport stream.

32. The method as claimed in claim 25, wherein the real-time seamless splicing is performed by a server when reading the first transport stream and the second transport stream from file storage and streaming the spliced transport stream to an application.

33. The method as claimed in claim 32, wherein the server streams the spliced transport stream to the application using a metered file transfer protocol (FTP).

34. The method as claimed in claim 32, which includes the server reading data of the first transport stream and the second transport stream from file storage and storing the data of the first transport stream and the second transport stream in a carousel of buffers, accessing the carousel of buffers to splice the first transport stream to the second transport stream and store data of the spliced transport stream in the carousel of buffers, and then accessing the carousel of buffers to transfer data of the spliced transport stream from the carousel of buffers for streaming to the application.

35. The method as claimed in claim 34, wherein the data of the first transport stream are stored in a first series of alternate buffers in sequence around the carousel, and data of the second transport stream are stored in a second series of alternate buffers in sequence around the carousel.

36. The method as claimed in claim 34, wherein the server includes a stream server computer and a cached disk storage subsystem, the carousel of buffers are allocated in random access memory of the stream server computer, and the stream server computer is programmed with a file system that maps the file storage to disk storage in the cached disk storage subsystem and that reads the data of the first transport stream and the second transport stream from disk storage of the cached disk storage subsystem for storing of the data of the first transport stream and the second transport stream in the carousel of buffers.

37. The method as claimed in claim 34, wherein the server includes a network interface board that reads the data of the spliced transport stream from the carousel of buffers for streaming to the application using a direct memory access protocol.

38. The method as claimed in claim 37, wherein the network interface board communicates with the application using a technology selected from group consisting of DVB, ATM, Ethernet, Fiber Channel, and Gigabit Ethernet.